МИНОБРНАУКИ РФ

ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №2 по дисциплине

"Логическое и функциональное программирование"

по теме «Создание сложной геометрической модели на базе функционального описания»

Вариант 12

Выполнил студент группы Б05-191-2: П. С. Костицын, Р.А. Гумметов

Принял: А. Г. Ложкин

Ижевск 2020

1. Постановка задачи

Написать функцию по построению трехмерного тела, состоящего из 3 элементарных. Элементарные фигуры располагаются вдоль оси Х. Схема расположения АВС. Если перед буквой стоит знак минус, то соответствующее тело убирается из тела В. Например, -АВ-С. Тела А и С вычитаются из В. В отличии от рисунка тела должны иметь широты и меридианы. Вывод фигур - функциями списковых циклов. На торцах широты и меридианы обязательны. Задавать количество вертексов. Проверять на физическую правильность построений. Размеры убираемого тела не должны превышать размера основного тела. Количество дополнительных функций не регламентируется. Запрещается использовать стандартные библиотеки. Тело строить из примитивов 3Dline или 3Dpoly.

2. Текст программы

(Defun c:drawFigure()

(InitGet (+ 1 8 16))

(setq startPoint (GETPOINT "Введите начальную точку\n\r"))

(InitGet (+ 1 2 4))

(setq hCyl (GETREAL "Введите высоту цилиндра\n\r"))

(InitGet (+ 1 2 4))

(setq rCylinder (GETREAL "Введите радиус цилиндра\n\r"))

(InitGet 7)

(setq hPyramid (GETREAL "Введите высоту усеченной пирамиды\n\r"))

(InitGet 7)

(setq rPyramid (GETREAL "Введите радиус усеченной пирамиды\n\r"))

(InitGet 7)

(setq hCone (GETREAL "Введите высоту усеченного конуса\n\r"))

(InitGet 7)

(setq rCone (GETREAL "Введите радиус усеченного конуса\n\r"))

(defun errorMessage()

(princ "Данные введены неверно")

(quit)

)

;Проверка корректности введенных данных

(if (OR (> rPyramid rCylinder) (> rCone rCylinder) (> hCone hCyl))

(errorMessage)

(princ)

)

(setq x (nth 0 startPoint))

(setq y (nth 1 startPoint))

(setq z (nth 2 startPoint))

;Переменные для цилиндра

(setq n 20)

(setq dt (/ (\* 2 pi) n))

(setq t1 dt)

(setq dx 4)

(setq xOffset 0)

(setq circleList (List (List (+ y rCylinder) z)))

;Список точек сечений цилиндра

(repeat n

(setq circleList (append circleList (List(List (+ y (\* rCylinder (cos t1))) (+ z (\* rCylinder (sin t1)))))))

(setq t1 (+ t1 dt))

)

;Меридианы цилиндра

(foreach point circleList (command "\_line" (cons x point) (cons (+ x hCyl) point) ""))

;Широты цилиндра

(repeat (+ (fix (/ hCyl dx)) 1)

(command "\_line" (cons (+ x xOffset)(car circleList)))

(foreach point circleList (command (cons (+ x xOffset) point)))

(command)

(setq xOffset(+ xOffset dx))

)

(Defun drawLine(pt1 pt2)

(command "\_line" pt1 pt2 "")

(princ)

)

(setq dtPyr (/ (\* 2 pi) 3))

(setq t1 dtPyr)

(setq pyramidBaseList (List (List (+ x hCyl) (+ y rPyramid) z)))

(setq nPyr 4)

;Точки для основания пирамиды

(repeat (- nPyr 1)

(setq pyramidBaseList (append pyramidBaseList (List(List (+ x hCyl) (+ y (\* rPyramid (cos t1))) (+ z (\* rPyramid (sin t1)))))))

(setq t1 (+ t1 dtPyr))

)

(setq dh 2)

(setq initialHPyr (+ 10 hPyramid))

(setq drPyramid (/ (\* dh rPyramid) initialHPyr))

(setq tempR rPyramid)

(setq tempH dh)

(command "\_line" (car pyramidBaseList))

(foreach point pyramidBaseList (command point))

(command)

;Широты пирамиды

(repeat (fix (/ hPyramid dh))

(setq t1 dtPyr)

(setq tempR (- tempR drPyramid))

(setq previousPoint (List (+ y tempR) z))

(repeat (- nPyr 1)

(setq currentPoint (List (+ y (\* tempR (cos t1))) (+ z (\* tempR (sin t1)))))

(command "\_line" (cons (+ x hCyl tempH) previousPoint) (cons (+ x hCyl tempH) currentPoint) "")

(setq t1 (+ t1 dtPyr))

(setq previousPoint currentPoint)

)

(setq tempH (+ tempH dh))

)

;Меридианы пирамиды

(setq tempH (- tempH dh))

(setq lastPyramidPoints (List (List (+ x hCyl tempH) (+ y tempR) z)))

(setq t1 dtPyr)

;Верхние точки пирамиды

(repeat (- nPyr 1)

(setq lastPyramidPoints (append lastPyramidPoints (List(List (+ x hCyl tempH) (+ y (\* tempR (cos t1))) (+ z (\* tempR (sin t1)))))))

(setq t1 (+ t1 dtPyr))

)

(mapcar 'drawLine pyramidBaseList lastPyramidPoints)

;Заполнение торца пирамиды

(foreach point lastPyramidPoints (command "\_line" point (List (+ x hCyl hPyramid) y z) ""))

(setq nCirclesOnPyramidFace 3)

(setq drPyramid (/ tempR nCirclesOnPyramidFace))

(setq tempR (- tempR drPyramid ))

(repeat (- nCirclesOnPyramidFace 1)

(setq t2 dtPyr)

(setq previousPoint (List (+ y tempR) z))

(repeat (- nPyr 1)

(setq currentPoint (List (+ y (\* tempR (cos t2))) (+ z (\* tempR (sin t2)))))

(command "\_line" (cons (+ x hCyl hPyramid) previousPoint) (cons (+ x hCyl hPyramid) currentPoint) "")

(setq t2 (+ t2 dtPyr))

(setq previousPoint currentPoint)

)

(setq tempR (- tempR drPyramid))

)

;Широты конуса

(setq dh 2)

(setq initialHCone (+ 10 hCone))

(setq drCone (/ (\* dh rCone) initialHCone))

(setq tempR rCone)

(setq tempH 0)

(setq dtCone dt)

(setq t2 dtCone)

(setq coneBaseList (List (List x (+ y rCone) z)))

(repeat (+ (fix (/ hCone dh)) 1)

(setq t2 dtCone)

(setq previousPoint (List (+ y tempR) z))

(repeat n

(setq currentPoint (List (+ y (\* tempR (cos t2))) (+ z (\* tempR (sin t2)))))

(command "\_line" (cons (+ x tempH) previousPoint) (cons (+ x tempH) currentPoint) "")

(setq t2 (+ t2 dtCone))

(setq previousPoint currentPoint)

)

(setq tempH (+ tempH dh))

(setq tempR (- tempR drCone))

)

;Меридианы конуса

(setq t2 dtCone)

(setq tempH (- tempH dh))

(setq tempR (+ tempR drCone))

(setq lastConePoints (List (List (+ x tempH) (+ y tempR) z)))

;Список с точками первой окружности конуса (R первой окр-ти = R конуса)

;Список с точками последней окружности конуса

(repeat n

(setq coneBaseList (append coneBaseList (List(List x (+ y (\* rCone (cos t2))) (+ z (\* rCone (sin t2)))))))

(setq lastConePoints (append lastConePoints (List(List (+ x tempH) (+ y (\* tempR (cos t2))) (+ z (\* tempR (sin t2)))))))

(setq t2 (+ t2 dtCone))

)

(mapcar 'drawLine coneBaseList lastConePoints)

;Соединение цилиндра с конусом

(setq num -1)

(foreach point circleList (command "\_line" (cons x point) (nth (setq num (1+ num)) coneBaseList) ""))

;(mapcar '(lambda ( pt1 pt2 ) (command "\_line" (cons x pt1) pt2 "")) circleList coneBaseList)

;Заполнение торца конуса

(foreach point lastConePoints (command "\_line" point (List (+ x hCone) y z) ""))

(setq nCirclesOnConeFace 3)

(setq drCone (/ tempR nCirclesOnConeFace))

(setq tempR (- tempR drCone ))

(repeat nCirclesOnConeFace

(setq t2 dtCone)

(setq previousPoint (List (+ y tempR) z))

(repeat n

(setq currentPoint (List (+ y (\* tempR (cos t2))) (+ z (\* tempR (sin t2)))))

(command "\_line" (cons (+ x hCone) previousPoint) (cons (+ x hCone) currentPoint) "")

(setq t2 (+ t2 dtCone))

(setq previousPoint currentPoint)

)

(setq tempR (- tempR drCone))

)

;Окружности на торце цилиндра со стороны конуса

(setq nCirclesOnFace 4)

(setq drFace (/ (- rCylinder rCone) nCirclesOnFace))

(setq tempR (- rCylinder drFace))

(repeat (- nCirclesOnFace 1)

(setq t2 dtCone)

(setq previousPoint (List (+ y tempR) z))

(repeat n

(setq currentPoint (List (+ y (\* tempR (cos t2))) (+ z (\* tempR (sin t2)))))

(command "\_line" (cons x previousPoint) (cons x currentPoint) "")

(setq t2 (+ t2 dtCone))

(setq previousPoint currentPoint)

)

(setq tempR (- tempR drFace))

)

;Окружности на торце цилиндра со стороны пирамиды

(setq drFace (/ (- rCylinder rPyramid) nCirclesOnFace))

(setq tempR (- rCylinder drFace))

(repeat nCirclesOnFace

(setq t2 dtCone)

(setq previousPoint (List (+ y tempR) z))

(repeat n

(setq currentPoint (List (+ y (\* tempR (cos t2))) (+ z (\* tempR (sin t2)))))

(command "\_line" (cons (+ x hCyl) previousPoint) (cons (+ x hCyl) currentPoint) "")

(setq t2 (+ t2 dtCone))

(setq previousPoint currentPoint)

)

(setq tempR (- tempR drFace))

)

;Соединение цилиндра с пирамидой

;(setq t2 dt)

;(setq cylFaceList (List (List (+ x hCyl) (+ y rPyramid) z)))

;(repeat n

; (setq cylFaceList (append cylFaceList (List(List (+ x hCyl) (+ y (\* rPyramid (cos t2))) (+ z (\* rPyramid (sin t2)))))))

; (setq t2 (+ t2 dt))

;)

;(setq num -1)

;(foreach point circleList (command "\_line" (cons (+ x hCyl) point) (nth (setq num (1+ num)) cylFaceList) ""))

(setq nCirclePoints (+ (fix (/ n (- nPyr 1))) 1))

(setq circlePointNum 0)

(setq circleCenter (List (+ x hCyl) y z))

(repeat nCirclePoints

(setq circlePoint (cons (+ x hCyl) (nth circlePointNum circleList)))

(setq intersPoint (inters circlePoint circleCenter (nth 0 pyramidBaseList) (nth 1 pyramidBaseList)))

(command "\_line" circlePoint intersPoint "")

(setq circlePointNum (+ circlePointNum 1))

)

(repeat nCirclePoints

(setq circlePoint (cons (+ x hCyl) (nth circlePointNum circleList)))

(setq intersPoint (inters circlePoint circleCenter (nth 1 pyramidBaseList) (nth 2 pyramidBaseList)))

(command "\_line" circlePoint intersPoint "")

(setq circlePointNum (+ circlePointNum 1))

)

(repeat nCirclePoints

(setq circlePoint (cons (+ x hCyl) (nth circlePointNum circleList)))

(setq intersPoint (inters circlePoint circleCenter (nth 2 pyramidBaseList) (nth 0 pyramidBaseList)))

(command "\_line" circlePoint intersPoint "")

(setq circlePointNum (+ circlePointNum 1))

)

(princ)

)

3. Результаты работы программы

Результат работы программы представлен на рис. 1 - рис. 4.

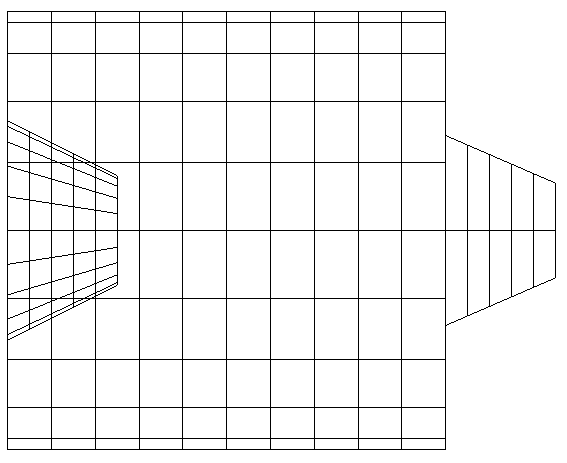


Рис. 1

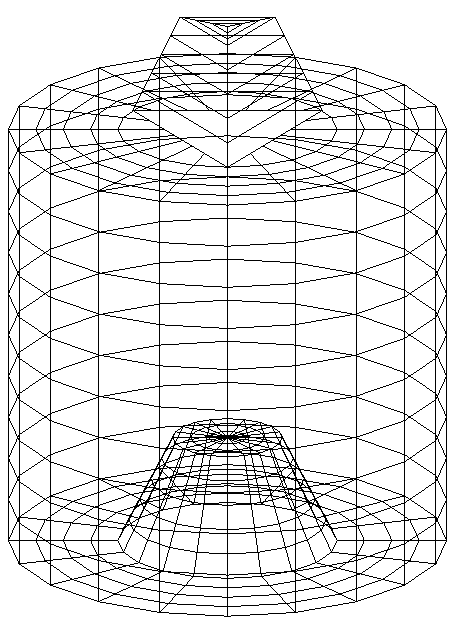


Рис. 2

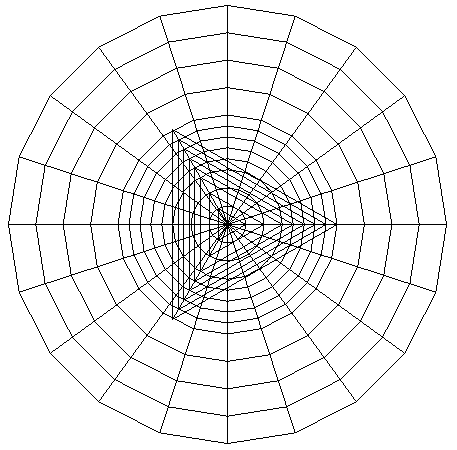


Рис. 3

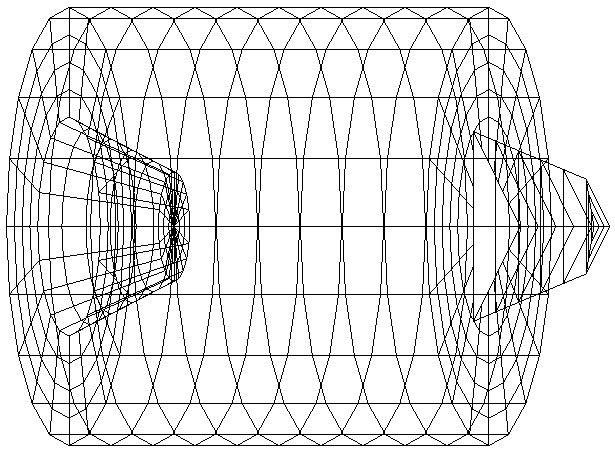


Рис. 4